Homework 3: Symbolic Planning

Rohan Tiwari

Andrew Id: rohantiw

Environments

Blocks and Triangles Environment

Conditions

- Clear(x) : x is Clear
- On(x,y) : x is on y
- Block(x) : x is Block
- Triangle(x) : x is Triangle
- NotTable(x) : x is not Table

Actions

- MoveToTable(b,x) : Move from x from b to Table
- Move(b,x,y) : Move b from x to y

Fire Extinguisher Environment

The symbols used are A,B,C,D,E are the possible start location, U (UAV), M (Mobile Robot), F1, F2, F3 (Fire Locations) and W (Water Location).

Conditions

- UAV(x) : x is UAV
- MobileRobot(x) : x is the Mobile Robot
- At(x,y) : x is at location y
- InAir(x) : x is the Air
- Fire(x) : x is Fire Location
- Water(x) : x is Water Location

- EmptyBattery(x) : x has Empty Battery
- EmptyTank(x) : x has Empty Tank
- ExtinguishedFire(x): Fire x is extinguished
- On(x,y) : x in on y

Actions

- Extinguish(x,y,z) : Extinguish Fire at location z where x is UAV and y is Mobile Robot
- Land(x,y,z) : Land x on y at location z
- Takeoff(x,y) : Takeoff x from y
- Move(x,y,z,w) : Move x and y to location z from location w
- MoveUGV(x,y,z,w) : Move from UGV y from w to z to meet with UAV x
- FillTank(x,y,z) : Fill Tank of the UAV x on UGV y at Water Location z

Results

Inadmissible Heuristic:

$h(s)$ = number of literals that are NOT yet satisfied i.e., h(s) = number of literals l_i such that $l_i(s) = false$ and $l_i(goal) = true$
I use weighted A^* with weight ϵ for weighted A^* : 10 for this heuristic.

Admissible Heuristic

For calculating the admissible heuristic I do a Dijkstra over the graph assuming actions don't have any negative effects. Then I use that as a heuristic for the Astar search.

No Heuristic

I am using Dijkstra search to get the shortest path.

Result on Given Environments

Fire Extinguisher Environment

Heuristic	Planning Time	No. of Steps	Expanded States
No Heuristic	7.91081 sec	19	138
Admissible	564.707 sec	19	100
Inadmissible	3.77573 sec	19	51

Blocks World Environment

Heuristic	Planning Time	No. of Steps	Expanded States
No Heuristic	0.0190805 sec	3	15
Admissible	0.18376 sec	3	4
Inadmissible	0.0158169 sec	3	5

Blocks and Triangles Environment

Heuristic	Planning Time	No. of Steps	Expanded States
No Heuristic	3.44047 sec	6	621
Admissible	429.778 sec	6	20
Inadmissible	0.521934 sec	7	82

Discussion of Result

From the above results we can observe that inadmissible heuristic is significantly faster especially in the case of the Fire Extinguisher environment. But the inadmissible heuristic gives an sub optimal path. In case of a simple Dijkstra search, we get an optimal path, but it is slower compared to the inadmissible heuristic and expands alot of states.The admissible heuristic gives an optimal path, but is significantly slower compared to both Dijkstra's search and the inadmissible heuristic, as we run a Breadth First Search over single node that is expanded and that makes its significantly slower.

Implementation Details

The pipeline are the following steps:

Generate Grounded Actions

To generate Grounded Actions, I get the number of arguments required by each action and then generate all possible permutations of the symbols. Then I generate all possible Grounded Actions by all combinations of symbols and Actions and save it in an *unordered_set*

Astar Search

I then find the path using Dijkstra's and Astar by using the implicit graph by considering all the possible actions on the current state and checking if the state satisfies the preconditions. Effects are applied to the current state to generate the new state. Edge cost for every node is taken as 1. One admissible and inadmissible heuristic is considered for Astar to guide the search.

How to Run

Run the command to Compile:

g++ -std=c++11 planner.cpp -o planner

Run Code:

./planner example.txt 1

0: Dijkstra Search(No Heuristic)
1: Inadmissible Heuristic
2: Admissible Heuristic